SEC.: [illegible] ROLL NO.: 200225014 0017

Serial No.	Date	Title

[SECTION - B]

{Answer - 6}

STACK - A stack is a linear DS in which we can say insert [PUSH] or delete [POP] an item at one end called the top of stack [TOS].

"STACK is based on LIFO (Last in first out) concept."

* PUSH - Insert new elements on the stack

→ Algorithm

```
PUSH (ITEM , TOP , SIZE , STACK)
Step-1:  if (TOP = = SIZE - 1) then
   (a)     print " stack is full"
   (b)     Exit .
Step-2:  else
           TOP = TOP + 1
Step-3:    STACK [TOP] = ITEM
Step-4:    Exit .
```

→ Cfunction -

```
void PUSH (){
   int int = 5;
   if (top = = size - 1)
   {
   printf(" stack is full ! ");
   getch();
   exit (0);
  }
   else {
   top = top +1;
   stack [top] = item;
        }
  }
```

* POP - Delete item from the stack.

→ Algorithm - POP(ITEM, STACK, TOP)

```
Step-1     If ( TOP == -1) then
  (a)      printf ("stack Empty").
  (b)      exit
Step-2   else
              ITEM = STACK [TOP]
Step 3         TOP = TOP - 1
Step 4         exit.
```

→ C function

```
        void POP(){
         int item = 1;
         if (top == -1)
         {
           printf (" Stack empty! ");
           getch();
           exit(0);
         }
         else{
           item = stack(top];
           top = top-1;
           printf (" The deleted item : %d", item);
          }
        }
```

[Answer – 7]

Algorithm to convert infix to postfix Notation:

Suppose Q is the infix Expression.
P = Postfix ~~Notati~~ expression.

1: PUSH '(' into stack & add ')' to the end of Q.

2: scanning from left to righ – repeat step 3 to 6 for each element of Q untill stack is empty.

3: if an operand is encountered add it to P

4: if a left paranthesis is encountered PUSH on stack.

-5: if an operator ⓧ is encountered then
ⓐ Repeatly POP from stack & add to P each operator on the TOS which has the same precedence or high precedence then ⓧ.
ⓑ Add ⓧ to stack.

-6: if a right paranthesis is encountered then
ⓐ Repeatly POP from stack.
ⓑ Remove the left paranthesis

p-7: exit

[Answer - 7(b)]

Q: A+(B*C-(D/E*F)*G)*H

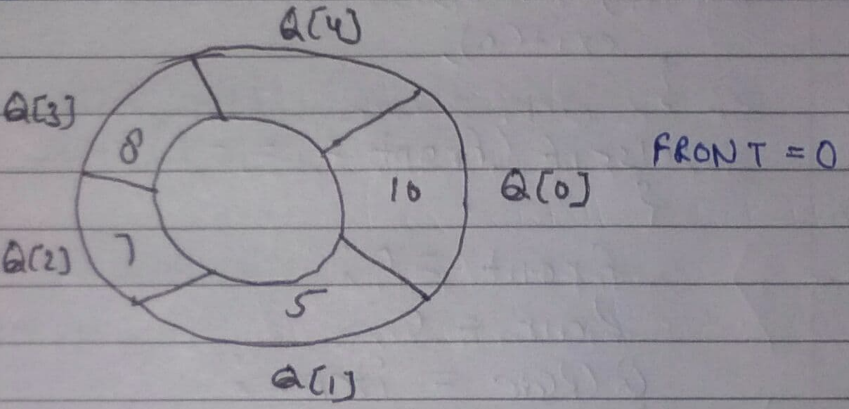
Symbol Scan	Stack	Postfix Expression
	(	
A	(	A
+	(+	A
(	(+(	A
B	(+(	AB
*	(+(*	AB
C	(+(*	ABC
-	(+(-	ABC*
(	(+(-(	ABC*
D	(+(-(	ABC*D
/	(+(-(/	ABC*D
E	(+(-(/	ABC*DE
*	(+(-(/*	ABC*DE
F	(+(-(/*	ABC*DEF
)	(+(-	ABC*DEF/*
*	(+(-*	ABC*DEF/*
G	(+(-*	ABC*DEF/*G
)	(+	ABC*DEF/*G-*
*	(+*	ABC*DEF/*G-*
H	(+*	ABC*DEF/*G-*H
)		ABC*DEF/*G-*H+*

$\boxed{ABC*DEF/*G-*H+*}$

[Answer-9]

CIRCULAR QUEUE

" It is a linear DS in which the operations are performed based on FIFO principle & the last position is connected back to the first position to make a circle "



Algorithm for insertion

```
if (front = (Rear +1) % Size), then
    print " Queue is full "
    Exit.

else if ( front == -1 ), then
    set front = 0, Rear = 0
    Q[Rear] = item

else
    Rear = (Rear +1) % Size
    Q[Rear] = item

STOP.
```

CFunction –

```
void CirQueueInsertion()
{
    int item = 8;
    if(front = (Rear +1) %. size)
    {
      printf(" Queue is full! ");
      getch();
      exit(0);
    }
    elseif (front = = -1)
    {
      front = 0;
      Rear = 0;
      Q(Rear = item;
    }
    else {
      Rear = (Rear +1) %. size;
      Q(Rear ) = item;
      }
    }
```

Algorithm for deletion.

```
1. if((front == -1) || (REAR == -1)) then
 (a) print " Queue is empty "
 (b) exit.
2. else if (front == Rear ) then
 (a) item = Q(front)
 (b) set front = -1, Rear = -1;
3. else      item = Q(front)
             front = (front +1) %. size
4. STOP.
```

C function –

```
void CirQueuedel()
{
  int item = 3;
  if ((Front == -1)||(Rear == -1))
   {
     printf("Queue empty");
     getch();
      exit(0);
   }
 else if (Front == Rear)
      {
        item = Q(Front);
         Front = -1;
         Rear = -1;
         printf("The deleted item:%d", item);
      }
  else
    {
       item = Q[Front];
       Front = [Front + 1] % size;
       printf("The deleted item = %d", item);
       }
   }
```

Answer -11 (a)

Priority Queue — It is an extension of Queue with following properties.

1- every item has a priority associated with it

2 - An element with high priority is dequeued before an element with low priority.

3- If 2 item have the same priority then are follow the concept of FIFO. Queue

Note- i) Insert (item, priority);
insert an item with given priority

ii) delete highest priority ( );
remove the highest priority item.

(b) Recursion - It is a technique by which a function call itself again & again until a loss condition is satisfied & that function is called recursion function.